



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/815,037	03/26/2004	Miroslav R. Petrov	6570P035	7811
45062	7590	04/09/2009	EXAMINER	
SAP/BSTZ			HASSAN, RASHEDUL	
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP			ART UNIT	PAPER NUMBER
1279 OAKMEAD PARKWAY				2179
SUNNYVALE, CA 94085-4040				
			MAIL DATE	DELIVERY MODE
			04/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/815,037	PETROV ET AL.	
	Examiner	Art Unit	
	RASHEDUL HASSAN	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 30 January 2009.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-3,5-13,15,16,20,21,23-26,28 and 29 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-3,5-13,15-16,20-21,23-26, and 28-29 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____. | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 01/30/2009 has been entered.

Claim Objections

Claim 10 is objected to because of the following informalities:

Claim 10 recites the limitation “*the central service having a locking service and a messaging service, , the locking service enabling synchronization...*” in line 19 (emphasis added). The extraneous “,” should be removed.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-3, 5-10, 13, 15-16, 20-21, 23-26, and 28-29 rejected under 35 U.S.C. 103(a) as being unpatentable over Hessmer et al. (US 2002/0112044 A1) hereinafter Hessmer, in view of E et al. (US 2004/0019639 A1) hereinafter E, and further in view of Melchione et al. (US 2002/0091819 A1) hereinafter Melchione.

For Claim 1, Hessmer teaches **a computer-implemented method employed within a network of application server instances** (e.g., one embodiment of a network of application server instances, e.g., data access server instances, is illustrated in Fig. 1, see [0028]) **comprising:**

displaying a hierarchical tree structure having one or more tree nodes in a graphical user interface, each of the one or more tree nodes representing a resource of an application server instance (i.e., see Fig. 4 illustrating a hierarchical tree structure having one or more tree nodes to be displayed in a left pane of a graphical user interface, each of the one or more tree nodes representing a resource of

an application server instance. See accompanying discussion in paragraphs [0056-0057]. Also see paragraphs [0024-0027] which provide description of a diagnostic object/root illustrated as tree nodes in Fig. 4).

The preamble of the claim further recites that the method is ***employed within a network of application server instances having a cluster architecture***, and the claim further recites that the above mentioned one or more tree nodes of the hierarchical tree structure represents a resource of an application server instance ***within a cluster of application server instances, each application server instance within the cluster of application server instances having***

a group of server nodes configured with a redundant set of application logic and associated data, each server node within the group of server nodes having access to a central database associated with the cluster of application server instances, and

a dispatcher in communication with a central service associated with the cluster of application server instances, the central service enabling synchronization and communication between each of the application server instances within the cluster of application server instance, the central service having a locking service and a messaging service, the locking service enabling synchronization by disabling access to a portion of configuration data and program code stored with the central database, the messaging service enabling communication among the groups of server nodes within each application server

instance within the cluster of application server instances using a message passing protocol.

Hessmer fails to teach employing his method (i.e., the method of displaying a hierarchical tree structure having one or more tree nodes in a graphical user interface, each of the one or more tree nodes representing a resource of an application server instance) within a network of application server instances having the cluster architecture with all of the above limitations recited in the claim. In particular, Hessmer fails to clearly address “*a redundant set of application logic and associated data*”, as well as “*access to a central database*”, “*a dispatcher*”, “*a central service*”, “*a locking service*” and “*a messaging service*” as recited in the claim. However, the Examiner notes that the instant specification does not provide any limiting definition for the terminology “dispatcher” utilized in the claim. Referring to Fig. 13, paragraph [00084] mentions, “In one embodiment, dispatcher 1312 distributes service requests from clients to one or more of server nodes 1314, 1316, 1318 based on the load on each of the servers”. Therefore, without limiting, one interpretation of the term “dispatcher” in light of one embodiment disclosed in the specification could be “a module used for load balancing”. The specification also does not provide any limiting definition for the terminology “locking service” and “messaging service” utilized in the claim. E teaches ***a cluster of application server instances*** (e.g., see Fig. 1 which shows a distributed data system including a cluster of application servers 104A and 104B which can facilitate distributed user sessions, wherein the limitation “cluster of application servers” is interpreted to mean a plurality of computers

inter-connected and grouped together in a network. Even if the limitation “cluster of application servers” is interpreted to require multiple server devices configured to provide the same service and even if the limitation is interpreted to imply resilience to failure and/or some kind of load balancing, E reference would still implicitly teach such limitations or at least suggests such limitations to one of ordinary skill in the art. For example, regarding distributed user sessions, E mentions, “[d]istributed sessions may be distributed among multiple servers, for example in a cluster, whereas local sessions may be bound to an individual server”. Emphasis added, see [0008]. In other words, E clearly implies or at least suggests that the application servers 104A and 104B can be organized in a cluster to handle distributed sessions), **each application server instance within the cluster of application server instances having**

a group of server nodes configured with a redundant set of application logic and associated data (e.g., each application server instance 104 within the cluster of application server instances as illustrated in Fig. 1 has a group of processes 106, and associated data 108. These processes 106 can be interpreted as the claimed “server nodes” since they provide data and/or services for use by the clients. See [0035], wherein E mentions, “In one embodiment, the applications and/or processes within the application servers may provide data and/or services to enterprise server 102, for example, for use by the clients”. In the alternative, even if the processes are, for the sake of argument, considered not server processes, it would have been at least obvious to those of ordinary skill in the art to have some of these processes as server processes as such

modification is considered not the result of innovation but of ordinary skill and common sense. Additionally, it is implicit or would have been at least obvious to those of ordinary skill in the art, according to the E reference that the processes 106 are, or can easily be, configured with a redundant set of application logic and associated data. This is because Fig. 1 illustrates a distributed data system, and regarding distributed data system, E explicitly mentions, "Distributed data systems may provide for load balancing and fail over to improve the overall quality of service of the system." See [0006]. Thus it is implicit or at least would have been obvious to have the distributed data system as illustrated in Fig. 1 to provide load balancing and fail-over mechanism and thereby providing redundant set of logic and associated data for some of the processes 106), **and**

a dispatcher (e.g., the module controlling the load balancing functionality) ***in communication with a central service associated with the cluster of application server instances*** (e.g., distributed store 110 in Fig. 1 can be interpreted as "a central service"), ***the central service enabling synchronization and communication between each of the application server instances within the cluster of application server instances, the central service having a locking service and a messaging service, the locking service enabling synchronization by disabling access to a portion of configuration data and program code stored within the central database, the messaging service enabling communication among the groups of server nodes within each application server instance within the cluster of application server instances using a message passing protocol*** (e.g., lock

mechanism 114 in Fig. 1. See [0038] and [0039] for locking portion of primary data 112 using messaging services for obtaining a token and thereby providing synchronization and communication between each of the application server instances within the cluster of application server instances for accessing and utilizing the primary data 112).

Therefore, E teaches the above additional limitations of the claim not explicitly taught in Hessmer. The question is whether it would have been obvious to one skilled in the art to combine the teaching of these two references. Hessmer teaches a method for collecting diagnostic data and presenting the diagnostic data in nodes of a GUI having a tree structure for monitoring the configuration and operation of data access servers and associated data source devices employed within a distributed data system (see Abstract, also [0006-0008], see Fig. 4 for the tree structure). His technique of providing a GUI with a tree structure wherein the nodes of the tree represent diagnostic data is similar to the claimed method and thus relied on by the Examiner in this rejection. Hessmer however does not explicitly mention using this GUI technique in a network having the cluster architecture as claimed. But, the Examiner considers that his method of presenting diagnostic information in a GUI according to the specified manner can obviously be used with many different network architectures, and in particular is very useful for distributed network architectures as explicitly suggested by Hessmer (see [0008]). E teaches an improved locking mechanism to help prevent inconsistency in the distributed data and avoid data clobbering issues that are often experienced in a distributed data system (see “*Description of the Related Art*” section). E illustrates his technique in a network having the clustered architecture similar to the one claimed. The

cluster architecture exemplified by E is a distributed network system for which Hessmer explicitly suggested using his technique in order to minimize the delays encountered when tuning the operation of the system as mentioned above. Therefore, the Examiner believes that it would have been obvious to a person of ordinary skill in the art to combine Hessmer and E in order to use the diagnostic GUI of Hessmer to monitor the clustered network used in E.

Claim 1 further requires, "***wherein at least one of the tree nodes represents a service of the application server instance within the cluster of application server instances***". The instant specification mentions, "The term 'service' refers to functionality derived from a particular software program. The functionality provided by a service may include, and is not limited to, lifecycle management, security, connectivity, transactions, and/or persistence" (see [00063]). Referring to Fig. 4 of Hessmer, some of the nodes of the hierarchical tree represents a "Root" of a particular type. Each of these Roots are software programs (see [0027]) that provides "functionality", i.e., provides monitoring information of a particular type associated with corresponding data access server (see Fig. 3 for various types of default diagnostic root types, i.e., service types. One of the root types is explicitly mentioned as "Transactions" 230). Thus this limitation is met by Hessmer.

Claim 1 further requires,
"receiving an input selecting the tree node representing the service of the application server instance within the cluster of application server instances;

displaying a list of one or more service references associated with the service represented by the selected tree node in the graphical user interface;

The instant specification mentions, “The term ‘service reference’ broadly refers to a software module that provides a ‘service’ to a service” (see [00065]). In other words, the limitations require that when a node that represents functionality of a software module or program is selected, a list of one or more software modules used by the software module represented by the node is displayed. It has been already discussed above that in Hessmer, the “Root” nodes in Fig. 4 represent a “service” or “functionality” of a software module, since “Root” is a “diagnostic object” created from a “class” and hence is a software module (see [0027]). Hessmer further teaches that once a Root is selected from the tree, lower levels and their associated information are further exposed (“scalability of elements to expose lower levels and their associated information”, see [0056]). Thus Hessmer teaches displaying a list of service references upon selection of a tree node representing service of the application server as required by the claim.

Claim 1 further requires, “***displaying a relationship value for each listed service reference, wherein the relationship value is to specify a strength of a relationship between the listed service reference and the service represented by the selected tree node***”. Although, Hessmer teaches displaying associated information of the service references, he nevertheless does not disclose “*displaying a relationship value for each listed service reference, wherein the relationship value is to specify a strength of a relationship between the listed service reference and the service*

represented by the selected tree node" as required by the claim. E also does not appear to cure this deficiency. The Examiner notes that dependent claims 5 and 6 that further limit the independent claim 1, further clarifies the scope of the limitation "*relationship value*" claimed in the independent claim. Based on the limitations of dependent claims 5 and 6, it appears that the "*relationship value*" is an item displayed in the GUI that indicates whether the listed service reference is to be automatically started or not when the service represented by the selected tree node is started. In other words, the "*strength of the relationship between the listed service reference and the service represented by the selected tree node*" is nothing more than an indication of whether the listed service reference is to be automatically started or not when the service represented by the selected tree node is started. This limitation would have been obvious over Hessmer and E, in view of Melchione. Similar to Hessmer, Melchione also teaches a method for the configuration, management, and/or monitoring of computer applications and devices via a computer network. Like Hessmer who teaches displaying a tree node representing a service (i.e., a Root/diagnostic object software module), Melchione also teaches at least one of the tree nodes represents a service (e.g., a tree node "VirusScan for Win9x" in Fig. 4 and 5) of a computer application and/or device, selecting the tree node representing the service (selection of "VirusScan For Win9x" as in Fig. 4 or "E-Mail scan" service as shown in Fig. 5), and displaying a list of one or more service references associated with the service represented by the selected tree node in the graphical user interface (multiple service references as shown in window pane 406B in Fig. 4 and 5), and displaying a relationship value, for each listed service

reference, wherein the relationship value is to specify whether the listed service reference is to be automatically started when the service represented by the selected tree node is started (e.g., the status of the radio buttons and check boxes constitute a relationship value that specify whether the associated service reference is to be automatically started when the service represented by the selected tree node is started, also see the prosecution history, specifically the “Response to Arguments” section of the OA on 07/16/2007 and the rejection of claim 1 in that OA). Therefore, Melchione illustrates an embodiment of one aspect of his invention that when combined with Hessmer and E teaches all the limitations of the instant invention. The Examiner believes that it would have been obvious to a person of ordinary skill in the art to combine Melchione with Hessmer and E in order to arrive at the present invention. For example, it would have been obvious to those skilled in the art to incorporate the services associated with the Virus scan service as illustrated by Melchione into the application servers of the combined invention of Hessmer and E because protecting the application servers using a virus scan service makes common sense and is considered to be highly desirable so that the servers can be protected from malicious software.

Independent claims 10 (apparatus), 20 (system) and 25 (an article of manufacture) recite similar limitations as claim 1 and therefore have been rejected under the same rationale as discussed in detail for claim 1.

For claim 2, Hessmer further teaches ***displaying the hierarchical tree structure having one or more tree nodes in the graphical user interface comprises:***

displaying the hierarchical tree structure in a first window pane of the graphical user interface (e.g., “In an embodiment of the invention, the diagnostic utility supports creating a graphical image in the form of a window comprising two side-by-side pages. A tree structure diagram having structures similar to the depicted structure of Fig. 4 is depicted in the left pane of the graphical user interface associated with the diagnostic utility 110. [0056]); ***and wherein displaying the list of one or more service references associated with the selected tree node in the graphical user interface comprises: displaying the list of one or more service references associated with the service represented by the selected tree node in a second window pane of the graphical user interface*** (e.g., When a user selects one of the depicted nodes of the depicted root structure, the diagnostic utility presents data related to the node, and currently possessed by the diagnostic utility, within the right side pane of the graphical user interface. [0056]).

Furthermore Melchione also teaches displaying the hierarchical tree structure having one or more tree nodes in the graphical user interface comprises:

displaying the hierarchical tree structure in a first window pane of the graphical user interface (404A in Fig. 4 and 404B in Fig. 5); and wherein displaying the list of one or more service references associated with the selected tree node in the graphical user interface comprises: displaying the list of one or more service references

associated with the service represented by the selected tree node in a second window pane of the graphical user interface (406A in Fig. 4 and 406B in Fig. 5).

For claims 3 (method), 13 (apparatus), 21 (system) and 26 (article of manufacture), Melchione further teaches ***displaying the list of one or more service references associated with the selected tree node comprises: displaying a service reference name, for each listed service reference, wherein the service reference name is to identify the service reference*** (“Enforce VirusScan for Win9x Policies”, “inherit” in Fig. 4; “Prompt for user action”, “Move infected files automatically”, “delete infected files automatically” etc. are considered as names for these service references as shown in Fig. 5).

For claims 5-6 (method), 15 (apparatus), 23 (system) and 28 (article of manufacture), Melchione further teaches ***wherein the displayed relationship value indicates that the strength of the relationship between the listed service reference and the service represented by the selected tree node is hard if the listed service reference is to be automatically started when the service represented by the selected tree node is started*** (e.g., selected status of the radio buttons and check boxes constitutes a hard relationship value); ***and the displayed relationship value indicates that the strength of the relationship between the listed service reference and the service represented by the selected tree node is weak if the listed service reference is not automatically started when the service***

represented by the selected tree node is started (e.g., unselected status of the radio buttons and check boxes constitutes a weak relationship value).

For claims 7-8 (method) and 29 (article of manufacture), Melchione further teaches ***displaying the list of one or more service references associated with the selected tree node further comprises: displaying a service reference type for each listed service reference, wherein the service reference type is to specify a service reference type for the listed service reference*** because displaying the name of each listed service reference also serves as displaying the type of the service reference. As shown if Fig. 4 and 5, the name of the depicted service references specifies the type of the service references as a “service” type.

For claims 9 (method), 16 (apparatus) and 24 (system), Melchione further teaches ***receiving an input selecting one of the listed service references; and setting the relationship value for the selected service reference, the relationship value to specify the strength of the relationship between the selected service reference and the service represented by the selected tree node, the strength of the relationship indicating whether the selected service reference is to be automatically started when the service represented by the selected tree node is started*** (e.g., pointing and clicking the mouse button to activate a radio button or check box constitutes selecting one of the listed service references and providing a relationship value). However, the Examiner notes that Melchione does not require

“receiving an additional input” for setting the relationship value for the selected service reference. In Melchione, selecting a radio button or check box achieves both selecting of the listed service reference and specifying the strength of the relationship using only one input. But such a minor difference in the mechanism of selecting and specifying a value for an option is not considered to constitute a novelty, but rather the result of ordinary skill and common sense. For example, the Examiner takes official notice that it was well known in the art at the time of the invention to use “tab” key of a key board to step through actionable items of a GUI for selecting one item and then use the “enter” key or the mouse “click” to provide or toggle between possible values for the item. Therefore it would have been obvious to provide such a selection mechanism in the proposed combination to require “an additional input” for setting the relationship value and thereby arrive at the present invention.

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hessmer, E, and Melchione as applied to claim 10 above, and further in view of Digiorgio et al. (US 2001/0005201) hereinafter Digiorgio.

For claim 11, Melchione does not teach that the graphical user interface is a Swing-based graphical user interface. However, the Java Swing technology was well known and widely used technology in the art for creating graphical user interface at the time of the invention. Digiorgio teaches displaying a GUI using Java Foundation Classes (JFC) that uses “Swing” ([0052]). Therefore, it would have been obvious for a

person of ordinary skill in the art at the time of the invention to modify Melchione's teaching with that of Digiorgio to utilize a Swing-based graphical user interface. The motivation would have been to achieve portability among various platforms and simplify implementation (Digiorgio, [0052]).

Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hessmer , E, and Melchione as applied to claim 10 above, and further in view of Ismael et al. (US 6,061,721) hereinafter Ismael.

For claim 12, Melchione does not teach that each of the one or more tree nodes comprises a managed bean to provide a management interface for the represented application server resource. Ismael teaches a bean-based management system that uses managed beans to abstract, control and monitor system resource using a graphical user interface. Therefore, it would have been obvious for a person of ordinary skill in the art at the time of the invention to modify Melchione's teaching with that of Ismael to use managed beans as tree nodes to provide a management interface for the represented application server resource. The motivation would have been to utilize the reusable component feature of a bean object and develop a more flexible network management environment (Ismael, column 2 lines 3-5).

Response to Arguments

Applicant's arguments filed 01/30/2009 have been fully considered but they are not persuasive.

In response to Applicants' argument that mere mention of the term "clustered" in describing the enterprise server 102 in E reference does not support the conclusion that E teaches the cluster of application server instances as recited in the claims (see page 17 in the Remarks), the Examiner would like to point out that Fig. 1 in E, shows a distributed data system including a cluster of application servers 104A and 104B to facilitate distributed user sessions, wherein the limitation "cluster of application servers" is interpreted to mean a plurality of computers inter-connected and grouped together in a network. Even if the limitation "cluster of application servers" is interpreted to require multiple server devices configured to provide the same service and even if the limitation is interpreted to imply resilience to failure and/or some kind of load balancing, E reference would still implicitly teach such limitations or at least suggests such limitations to one of ordinary skill in the art. For example, regarding distributed user sessions, E mentions, "[d]istributed sessions may be distributed among multiple servers, for example in a cluster, whereas local sessions may be bound to an individual server". Emphasis added, see [0008]. E further mentions, "Distributed data systems may provide for load balancing and fail over to improve the overall quality of service of the system." See [0006]. In other words, E clearly implies or at least suggests that the application servers 104A and 104B can be organized in a cluster (i.e., even if the limitation is interpreted to imply resilience to failure and/or some kind of load balancing) to handle distributed sessions.

In response to Applicants' argument, "a review of the E reference reveals that a process 106 executes within an application server 104, may be multithreaded, and may include a virtual machine (E, paragraph [0036]-[0037], all of which would indicate that, although the processes 106 in E are executing software, they are not server nodes as that term is described and claimed in the present application" (see page 18 in the Remarks), the Examiner points out that without any limiting definition within the specification, the term "server" is interpreted according to the broadest reasonable interpretation as "a program which provides some service to other (client) programs" (see the definition of "server" in Free Online Dictionary of Computing: <http://foldoc.org/index.cgi?query=server&action=Search>). E clearly mentions, "In one embodiment, the applications and/or processes within the application servers may provide data and/or services to enterprise server 102, for example, for use by the clients". Emphasis added, see [0035]. Therefore, the Examiner considers that the processes 106 can reasonably be interpreted as "server" processes, or at least it would have been obvious to one of ordinary skill in the art to employ "server" processes for some of the processes 106 mentioned by E since such modification is considered not the result of innovation but of ordinary skill and common sense.

In response to Applicants' argument that there is nothing in the disclosure of Melchione or the other cited references of record that discloses specifying a strength of a relationship as recited in the claims (see page 18 in the Remarks), the Examiner

points out that dependent claims 5 and 6 further limit the independent claim 1, further clarifies the scope of the limitation "*relationship value*" claimed in the independent claim. Based on the limitations of dependent claims 5 and 6, it appears that the "*relationship value*" is an item displayed in the GUI that indicates whether the listed service reference is to be automatically started or not when the service represented by the selected tree node is started. In other words, the "*strength of the relationship between the listed service reference and the service represented by the selected tree node*" is nothing more than an indication of whether the listed service reference is to be automatically started or not when the service represented by the selected tree node is started. Melchione teaches this since the status of the radio buttons and check boxes (see window 406B in Fig. 4 and Fig. 5) constitute a relationship value that specify whether the associated service reference is to be automatically started when the service represented by the selected tree node is started.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RASHEDUL HASSAN whose telephone number is (571)272-9481. The examiner can normally be reached on M-F 7:30AM - 4PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rashedul Hassan/
Examiner, Art Unit 2179

/Weilun Lo/
Supervisory Patent Examiner, Art Unit 2179